

REMARKS

Claims 1-35 have been cancelled and claims 36-46 to more clearly recite the invention. As explained in more detail below, support for the amendments is found in Fig. 2 and in the description in paragraph [0019] of the original specification. It is respectfully submitted that once the Examiner appreciates how simple the basic invention really is¹, as discussed below, the Examiner will appreciate that the amendments in fact are fully supported by the original specification and do not involve “new matter.”

Reconsideration of the application is respectfully requested for the following reasons:

1. Rejection of Claims 24-35 USC §112, 1st Paragraph

This rejection is respectfully traversed on the grounds that the invention as now claimed could easily be implemented by those skilled in the art since it simply involves storing extension header information in a “cache” (*i.e.*, temporary memory) and reading the cached headers in parallel whenever packets with the same header information are encountered. The invention is not a “caching” method, but rather merely uses a cache to store data that can speed up processing of identical packets.

The invention operates as follows:

- The header of a packet is checked to see if there are extension headers. The information that indicates whether there are extension headers is used as a key.
- The first time the “key” is encountered, there can be no corresponding cache data. Therefore, the extension headers are read in serial fashion in the conventional manner. The headers are “cached” or stored as they are read in the usual manner. The reason is that packets with the same extension headers may be encountered later.

¹ To summarize: (I) read extension headers in conventional serial fashion → (ii) store the read data (such as extension header lengths) in a cache → (iii) use the stored data to read the extension headers in parallel the next time an identical packet (one with the same “key”) is received

Serial Number 10/691,515

- The next time a “key” is encountered, the “key” is used to access the previously cached data. The stored (“cached”) extension headers can be loaded in parallel because they have already been read one time and therefore the lengths of the extension headers are known. Because the stored extension headers are loaded in parallel, processing time is reduced.

That is the basic invention—read serially in conventional fashion, store, and retrieve. Each of the steps involved, including reading of extension headers, storing data in a cache, and retrieving the data based on a key made of header fields, is easily within the ability of those skilled in the art.

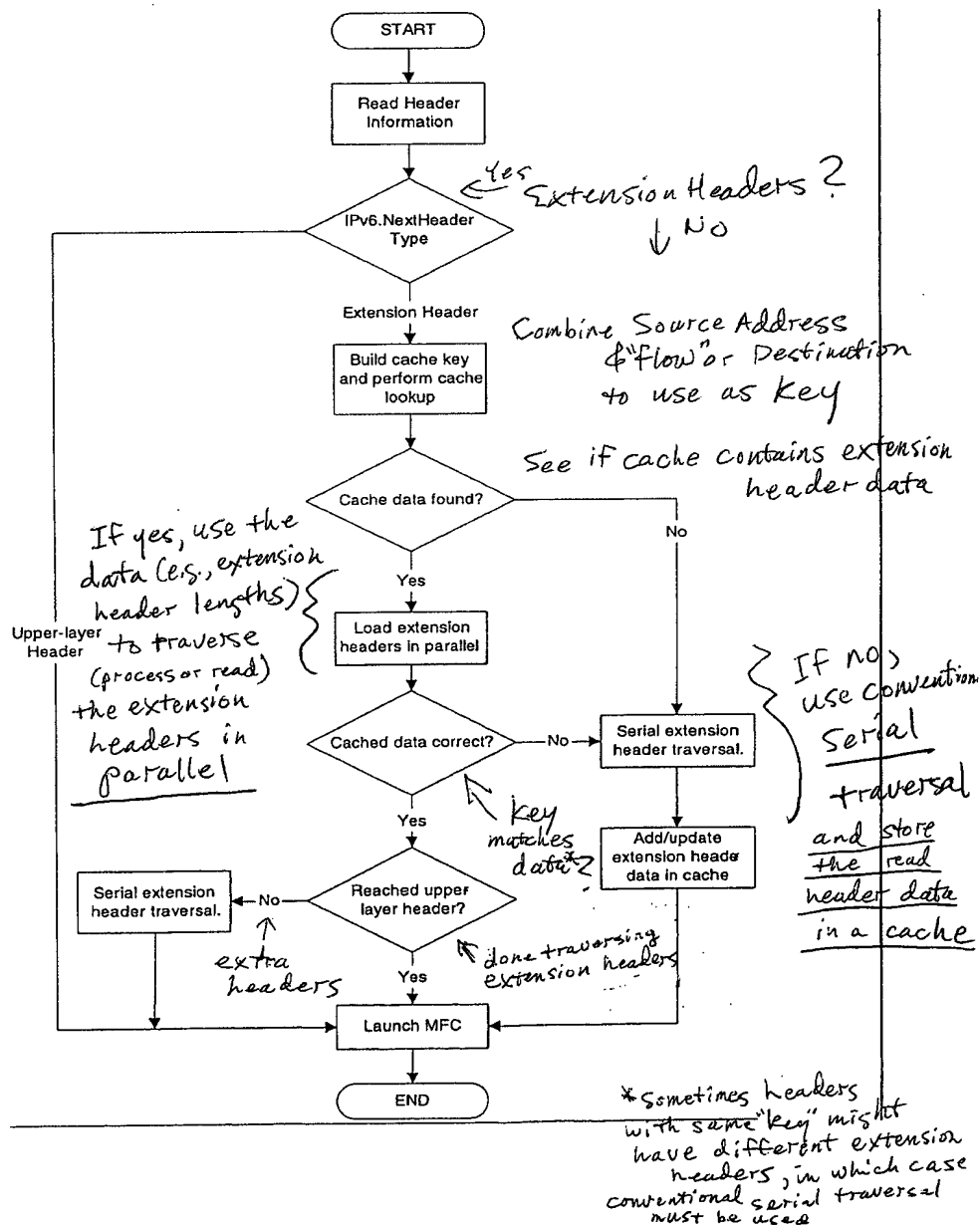
There are some refinements, such as the specific header information used as the cache “key” (such as IP source address and flow label or IP source and destination addresses) and performing serial extension header traversal when the cache data is completely read and the upper header has not yet been reached, but the basic principle of the invention boils down to:

- read header
- extract key from header fields (such as source address and flow label)
- see if there is stored (“cache”) data corresponding to the key
- if no, reading extension headers (if any) in conventional serial fashion and store data concerning the extension headers (such as header lengths) in a cache
- if yes, using the extension header data in the cache to read the extension headers in parallel, thereby expediting processing of the extension headers.

Of course, if extension header data in the cache does not match the actual extension headers in the packet, then the cache cannot be used and the extension headers must be read in conventional serial fashion. However, many packets will in fact have the same extension headers if they are from the same source and have the same flow (or destination address), so for those packets the invention can result in significant savings in processing time.

Upon reviewing the previous claims, it is understandable that the Examiner might have thought that the invention is much more complicated than it actually is. However, the claims have now been re-written to more clearly recite the invention. Description of the claimed steps

is found, for example, in paragraph [0019]. For example, the step of serial traversal of the extension headers if no entry is found is described in lines 19-21 on page 5. Placement of extension header data in a cache as a result of the serial traversal is found in lines 21-24 of page 5. Parallel traversal of the extension headers if cached data is found is described in lines 24-29 of page 5. In addition, these steps are illustrated in Fig. 2. For the Examiner's convenience, I have added annotations to the flowchart of Fig. 2 (which are based on the description in paragraph [0019]) in order to assist the Examiner in understanding the invention, as follows:



Serial Number 10/691,515

Although it is believed that the above should be sufficient to demonstrate that the invention can easily be implemented by those skilled in the art, some of the Examiner's questions indicate that the Examiner may not quite understand the basic concept of a "cache." About.com, for example, includes the following definition:

Definition: A cache is a block of

1. **RAM** used for temporary storage of data that is likely to be used again. The **CPU** and hard drive frequently use a cache, as do web browsers.

In a CPU there can be several caches, to speed up instructions in loops or to store often accessed data. These caches are small but very fast. Reading data from cache memory is much faster than reading it from RAM.

Similarly, Wikipedia includes the following definition of "cache":

1. **In computer science**, a **cache** (pronounced /kæʃ?/, like "cash" [1]) is a collection of data duplicating original values stored elsewhere or computed earlier, where the original data is expensive to fetch (owing to longer access time) or to compute, compared to the cost of reading the cache. In other words, a cache is a temporary storage area where frequently accessed data can be stored for rapid access. Once the data is stored in the cache, future use can be made by accessing the cached copy rather than re-fetching or recomputing the original data, so that the average access time is shorter. A cache, therefore, helps expedite data access that the CPU would otherwise need to fetch from main memory.

It can be understood from these definitions that a cache is nothing more than a place to store data, in this case data concerning the lengths of extension headers. By storing the extension header data in a cache, the header data can be accessed without having to serially traverse all of the extension headers to extract the lengths, thereby saving time. In other words, a cache is simply a data store for quick access, which is exactly the purpose of the claimed cache.

Once the invention is understood to involve storing extension data in a "cache" to save time in processing the next packet with the same header data, the Examiner's specific objections set forth on page 5 are easily answered. According to the Examiner "Applicant failed to disclose the location of the caching of data." However, it does not actually matter where the data is "cached" or stored, so long as a key is provided, the key being specifically

described in the specification as the source address and flow label or destination address (each of which is found in a conventional packet header). It is normal for data to be cached at arbitrary absolute locations in a RAM, with access being provided by a key or pointer rather than by absolute “location.” As a result, it is respectfully submitted that the original specification in fact provides sufficient information to “locate” the cache.

Similarly, the phrase “to arrive at and read fields in the upper-layer header” simply refers to the fact that, when all of the extension headers have been read, the packet processor proceeds to read the upper header of the of the packet, which is not only within the capabilities of those skilled in the art, but in fact is the conventional way of proceeding.

Finally, with respect to the recitation of lacking an Internet Header Length field, it is noted that this recitation has been deleted. A packet with an Internet Header Length (IHL) field will simply be processed in conventional IPv4 fashion packets that lack IHL also lack extension headers (IHL is used to indicate data within the IPv4 header that is moved to extension headers in IPv6). In fact, it is the lack of IHL that makes the invention useful for IPv6 packets, since the length information is stored in individual extension headers rather than the IHL field, making serial traversal necessary. Even the present invention needs to use serial traversal the first time a particular type of packet is encountered, with the caching being used to speed-up processing the *next time* the same identical type of packet is encountered.

In view of the above explanation, and the amendments to the claims, which are fully supported by the original specification, withdrawal of the rejection under 35 USC §112, 1st Paragraph, and early and favorable action on the merits, is respectfully requested.

Serial Number 10/691,515

Having thus overcome each of the rejections made in the Official Action, withdrawal of the rejections and expedited passage of the application to issue is requested.

Respectfully submitted,

BACON & THOMAS, PLLC

A handwritten signature in black ink, consisting of a stylized 'B' followed by a long horizontal line.

By: BENJAMIN E. URCIA
Registration No. 33,805

Date: September 25, 2008

BACON & THOMAS, PLLC
625 Slaters Lane, 4th Floor
Alexandria, Virginia 22314

Telephone: (703) 683-0500

NWB:S:\Producer\ben\Pending Q...Z\W\WILSON 691515u03.wpd